5. KEY PLAYERS AND RELATIONSHIPS

This section provides:

A description of each of the “key players,” or types of personnel involved in GCCS architecture

Guidelines for establishing productive relationships among these players.

CONVERGENCE

GCCS is about all kinds of convergence: Joint Service requirements, shared battlefield information, and the meshing of text, graphics, audio, video, geodesy, and other kinds of data to provide the best response to warfighter needs. The technical disciplines underlying GCCS are also converging. Communications and information systems are no longer separable, as is best exemplified by the old Defense Communications Agency’s rebirth a few years ago as DISA.

All of this convergence has significant implications for the personnel involved in GCCS. You may feel some pressure to be an architecture specialist, an engineer, a planner for the future, a manager, a marketer, and a genius with graphics – sometimes simultaneously. To be effective, individuals must not only dip into many different skill areas, but must also work together with people at all levels and in all functional areas of the GCCS community.

This section will help you identify the people who depend on your support and to whom you can turn for help. It may also help you to better understand your role in the larger context of GCCS and to capitalize on your position and your abilities.

WHO’S WHO

Throughout, this document has identified several categories of people that are involved in developing or using GCCS architecture. The relationships of these personnel are illustrated, in an obviously simplified and generalized form, in Figure 5-1. Each one isdetailed in the following sections.

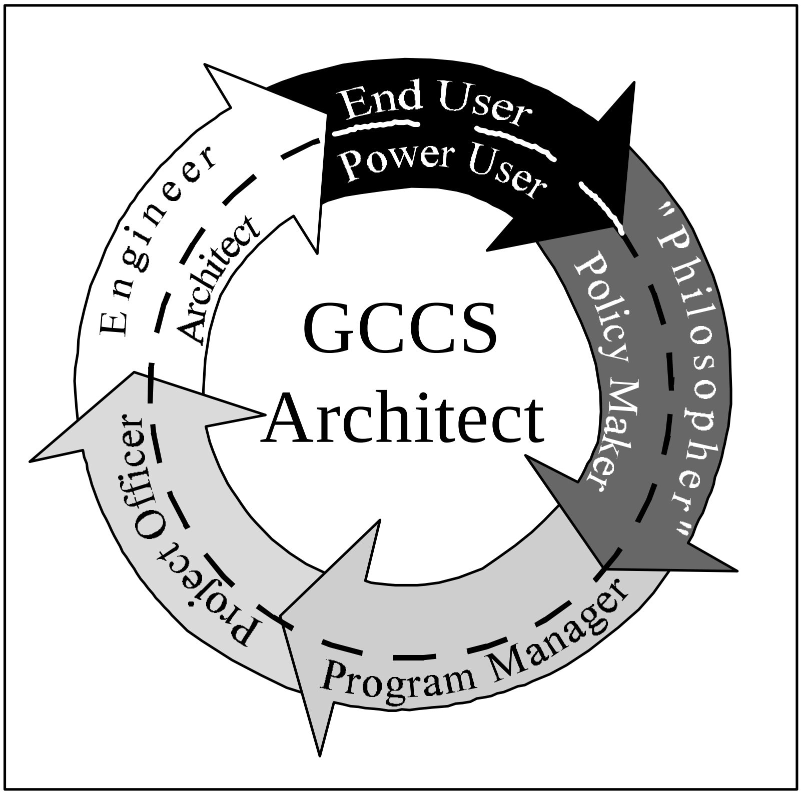

Figure 5-1.
Relationships of GCCS Architecture Key Players

The GCCS Architect

As shown in the illustration, the GCCS Architect is the central figure in the process of developing, maintaining, and acting upon GCCS architectures. The GCCS Architect is located at the DISA headquarters. Responsibilities of the GCCS Architect include:

- Developing, updating, and disseminating the overall GCCS architecture
- Developing template architectures to assist users in adapting the GCCS standards to individual sites
- Developing architecture conventions and methodology for GCCS, as contained in this document
- Reviewing drawings of record, drawings of acceptance, and other materials generated at GCCS sites
- Maintaining the GCCS architecture archive, as described in Section 4
- Resolving misunderstandings arising from erroneous interpretation or application of GCCS architecture guidelines.

The GCCS Architect – a singular term – is distinct from architects, plural. The GCCS architect provides overarching, binding guidance and maintains the official GCCS architecture repository for DoD. Information system architects may reside at individual GCCS sites. Their function is to prepare and maintain architecture data for that site and to ensure that the correct information is forwarded to the GCCS architect.

The End User

GCCS is the product of a strengthened focus on the information needs of the warfighter, who is the ultimate end user. The end user’s requirements are the catalyst for the activities performed by all the other players.

While it is available to anyone, the architecture guidance in this document is targeted toward two particular types of GCCS end users. The first is the site manager, who needs to know the configuration of all GCCS equipment at a site, understand how it meshes with local systems, and keep it operational.

The second is the end user who, by virtue of personal ability and organizational necessity, becomes the unofficial local expert on a system, network, or piece of equipment. This high-tech “power user” has a detailed working knowledge of how things work, what connects where, and how to keep it all running and get the job done when traditional knowledge resources are not available.

The site manager and local expert have a strong need for architecture information to use as a reference when troubleshooting, implementing upgrades, and proposing changes at the local level that need to be in consonance with the larger GCCS architecture. These personnel can also be valuable sources of detailed local information to be tapped by engineers and architecture experts developing or updating low-level architectures.

The “Philosopher” or Policy-Maker

With the current emphasis on Joint solutions to common warfighter requirements, efforts to satisfy those requirements typically begin at a high level in the defense hierarchy. High-level “philosophers” and policy-makers have the perspective to distill a range of end-user requirements into a set of core objectives, and to envision how those objectives can be met in the existing defense environment. They may sketch out the broad outlines of a concept – the “infosphere” at the heart of C4I for the Warrior is an example – and then task information system architects and engineers to fill in the details.

Philosophers are likely to be creators, or at least primary motivators, of strategic architectures, representational architectures, and other high-level, generalized depictions. They may also lead the development of target architectures, with significant input from experts who can forecast technological capabilities for the target period.

The Program Manager

The GCCS Program Manager is responsible for all aspects of GCCS development. The Program Manager is primarily a user of architecture information. The Program Manager directs subordinate staff members to prepare architecture information in a format that supports managerial review, reporting, and coordination. The GCCS Program Manager typically requires high-level or summary data, such as briefing slides showing current program status and targeted activity tied to specific milestones.

The GCCS Program Manager directs the activity of a number of project officers who have responsibility for some facet of GCCS development, such as the migration of stovepipe systems, software development, etc.

The Project Officer

The project officer heads a defined, bounded task either in the central GCCS program office or at a specific site. While “GCCS Program Manager” is a formal title, often “project officer” is not. Instead, someone may be informally appointed to coordinate a LAN upgrade or oversee the installation of new printers at a site.

While the project officer's responsibilities are typically narrow in scope, the project officer needs to have insight into the larger GCCS architecture to know what the constraints on activity may be. For example, the GCCS architecture may mandate that certain servers are dedicated to one function, or that a particular network configuration cannot be altered. The project officer will also use more detailed architecture depictions to plan and guide task activity, and will provide updates to those architectures as changes are implemented.

The Architecture Specialist

As noted in the discussion of the GCCS Architect’s role, an organization or local GCCS site may or may not have an architecture specialist. In an organization with abundant staff resources, there may be an employee charged with conducting business process redesign, local architecture development, maintenance of an architecture repository, and other tasks. Alternatively, a contractor may perform these services.

In more constrained circumstances, architecture work may be handled by an engineer, either as a formal responsibility or on an as-needed basis. In some cases, a person in charge of developing briefings may be appointed as the “architecture specialist,” since automated tools automate the work to a degree and production of architecture depictions for presentations is a near-constant activity.

The architecture specialist needs to retain copies of all architectures, both formal drawings of agreement and unofficial working drawings that are produced at a site. The architecture specialists needs to ensure that all drawings of agreement are forwarded to the GCCS architect, and that updates are generated and submitted as needed. The architecture specialist needs to play a key role in interpreting and ensuring compliance with architecture guidance that is received from the GCCS architect.

Optimally, the architecture specialist should also be familiar with information technology standards. The architecture specialist can advise the engineer on technologies that are compliant with both GCCS and DoD standards guidelines, such as the TAFIM.

The Engineer

GCCS is the handiwork of both the GCCS Engineer, who works closely with the GCCS Architect, and a number of engineers at the various GCCS sites.

The engineer's responsibilities include both executing the plans embodied in the GCCS architecture and assisting in shaping those plans. The engineer is responsible for delivering operating capabilities to the end user. The engineer also feeds into the architecture process by translating end user operational needs (e.g., "I need access to a color printer from this PC!") into workable engineering plans (e.g., a modified network routing scheme to make the needed link between PC and printer).

At a higher level, the GCCS Engineer guides the work of the GCCS Architect. The GCCS Engineer provides insight into "best of breed" technologies and highlights potential problems or risk factors due to maintainability, technological immaturity, noncompliance with standards, or other issues. The activities of the GCCS Engineer relative to those of the GCCS Architect are addressed in detail later in this section.

Engineers need to be familiar with architectures at all levels of the GCCS infrastructure from top-level strategic plans to detailed site baselines and installation drawings. Engineers will contribute to and use most types of architecture depictions on a frequent basis.

PATTERNS OF ARCHITECTURE USAGE

Everyone involved in GCCS has distinct requirements for architecture depictions, as discussed in the preceding sections. These requirements can be summarized by characterizing three levels of architecture usage:

- Creation/control of an architecture
- Contribution to an architecture
- Use of the end product.

To develop a clearer frame of reference for prioritization and decision-making about architecture work, consider who in your organization creates architectures, who contributes to them, and who is the primary or ultimate consumer for each one. Some key questions to ask include:

- For the architectures I create, do I have a close working relationship with the people who should be contributing to these drawings? Who else can I call upon?
- Are the resources we spend on architecture development in line with the needs of the architecture consumers? Do we provide adequate and accurate information for decision-makers and different, though equally important, data for implementation personnel?

- Can any operational problems or misunderstandings be avoided by bringing additional people "into the loop," or by changing the way we present architecture information?

THE ARCHITECT-ENGINEER PARTNERSHIP

A critical relationship is that between the architect and engineer. As is discussed in Appendix A, systems architecture has become an essential element of information systems development, but architecture and systems engineering have somewhat different roots. Engineering has typically focused on the development of an end product, using only as much of a conceptual framework as is necessary to get the engineer from Point A to Point B. Architecture has traditionally been allied with more academic or analytical practices, such as requirements analysis and business process redesign. Thus, while the growing complexity of information systems is pushing the engineer and the architect to become closer in their skills and activities, each one may approach a task from a distinct perspective and bring different intellectual assets to the job.

The GCCS Architect and the GCCS Engineer work in tandem to design the global infrastructure for GCCS and its site-specific components. Which one is in charge? Neither one. The GCCS Architect may initiate the design process by developing an architecture – a template architecture for configuring GCCS components with typical legacy systems, for example – but before it can be released, the GCCS Engineer must review the plan and ensure its technical validity and applicability to user sites. Conversely, the GCCS Engineer may develop an approach to applying emerging technology – such as a new information systems security capability – to simplify and improve the GCCS architecture. Before this capability can be incorporated, the GCCS Architect needs to assess its impacts throughout the global, multilayered GCCS architecture.

The same sort of architect-engineer partnership prevails at the local level. The architecture specialist needs to ensure that the drawings of record and supporting site data are accurate and complete. Ongoing contact with the engineer, or with multiple engineers serving as project officers, and even with knowledgeable end users is the best way to accomplish this. The engineer not only provides current information to the architecture specialist, but also must rely on the architecture specialist to preserve the "corporate history" of the organization. This aspect is especially important in military organizations where personnel turnover is high. The actual person performing architecture work may change just as the engineering staff may, but the architectural records, if properly maintained, are a constant, reliable reference.

Are You Either, Neither, or All of the Above?

What if you don't fit neatly into any of the categories described thus far? Maybe you have responsibilities that span a couple of categories, or maybe you come from a very different background and have landed in your present position unexpectedly. If this describes you, here are some suggestions:

- Remember the trend toward convergence. People who can see beyond organizational or technical boundaries are especially valuable in this environment.
- Follow the GCCS example. As GCCS incorporates "best of breed" technology, look for ways to combine the best practices from various disciplines, or to apply the experience you've gained in other settings.
- Look for automated tools to make your job easier. Section 6 and Appendix C of this document can help you identify appropriate kinds of tools.
- Call for help when you need it. The GCCS Architect at the DISA is a good starting point.
- Cultivate others like you. Look for smart individuals in your organization that can assist with updating architectures, archiving drawings, or maintaining the lines of communication.

6. TOOLS IN ARCHITECTURE DEVELOPMENT AND DOCUMENTATION

This section provides:

An analysis of the need for automated tools to support GCCS architecture development

A description of how the different developers and users of GCCS can use the various tools

Common tool requirements to support GCCS architectural development.

WHY AUTOMATED TOOLS?

The need for automation support tools for network designers and architects is important for many reasons. Consider the following scenario:

A major upgrade to a network is underway. Vendor solicitations are out, seeking information and availability of products and technology. How do you rapidly portray the planned upgrade so that all vendors can "see" it and can reference a common framework? Do you use words or a briefing slide? Can the vendors understand what you are trying to accomplish? Can your management understand this upgrade in nontechnical terms? What does a resegment of the network with a FDDI backbone administrative ring mean to a manager or to acquisition support personnel? Can it be portrayed graphically, showing "here it is today" and "this is what it will look like after the upgrade" (as in Figure 6-1)?

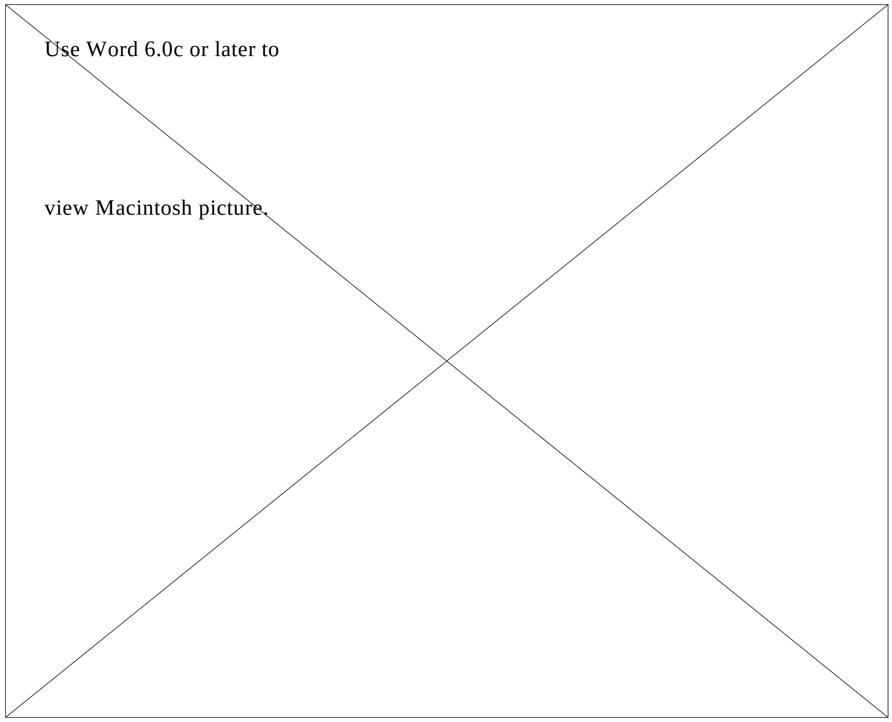

Figure 6-1.
Example Network

Questions such as these often confront network planners and designers . Automation support tools provide insight into three major categories of support:

- Documentation
- Accounting and reporting
- Modeling and simulation.

These three aspects are important to everyone involved in the design, acquisition, management, or operation of a client/server architecture.

Documentation

How many times have you had to draw the same thing over and over again, or provide a quick briefing on the current state or planned system upgrades? Architecture development is an iterative process that begins with the initial design and implementation, continues with performance improvements, and follows with future planning and upgrades. The steps of this process are interrelated, from the basic understanding of the initial design to completion of the final design. Many tools have emerged that help architects, planners, and engineers efficiently design, test, and track this process. Automated tools have become a vital part of the architecture development process. Any time a network design is initiated, the first step you go through is to physically represent the network with a diagram. This diagram provides a common view and framework for all involved personnel to analyze and agree upon. A typical physical view of the architecture depicts the location of servers, major networking components, and cabling types to help designers, planners, and implementors with future decisions.

Automated network design tools are becoming as important to network designers, as financial spreadsheets are to financial analysts. The tools provide increased productivity, accuracy, and presentation capabilities. The complexities and size of networks have vastly increased in the last few years, making the operation and maintenance of manual drawings particularly burdensome. Use of automated tools with a layering capability can greatly reduce the time and manpower required to properly lay out a physical representation of the network.

Automated tools are necessary today just to document and present LAN and WAN topology in an economical fashion. The sheer size and complexity of the architecture for GCCS would make it impossible to manage networks using a manual process. Automated tools make it easy to portray the architecture in a graphical fashion, and to document information on related components, applications, and processes related to physical components as well.

At the baseline level, managers, administrators, and end users are also concerned with the in-place network. Again, a physical representation of the network is the best way to provide a common framework. Documentation of the existing architectures is important to help administrators, engineers, and planners determine the baseline, assist in troubleshooting, and to help plan future upgrades appropriate to the network. For example, suppose one of a network's primary servers is no longer available and it is late at night. Help will not be available for several hours, but certain information is urgently needed. Where does the information reside? Is the server up and running? Is the network path established? Could a connection be down and/or broken? As an operator, you could use the baseline and technical architecture to help troubleshoot and answer these questions. These drawings would be particularly valuable if you were a new user with a technical background. Examination of an architectural diagram can help pinpoint congestion, chokepoints, and possible interface problems.

Developing physical layouts for large networks can be both time-consuming and expensive, particularly for large-scale enterprise networks such as GCCS. Automated tools can again assist administrators and planners by providing the ability to automatically query network devices, plot the configuration, and layer the view towards enterprise-wide WAN or a building LAN wiring diagram. The automatic plots can also be integrated with new drawing tools to help portray the system architecture for the present configuration, with layers to address all planned upgrades.

Accounting and Reporting

Sometimes you have to ask yourself if the equipment grew legs overnight. You come in and the workstation, router, or server has been moved. Changes always happen – it's a fact of life.

How do you keep up with these constant changes? Automated tools help track the changes that can occur overnight. Not only can you portray your network and systems automatically and in compact briefings, but additional information can also be pulled from these tools.

How many SunSPARC computers do we have on the fourth floor in Operations? What configuration files are maintained on the logistics servers? Automated tools for architectures can maintain this information within their databases. These tools provide a link to databases that can store hundreds of pieces of information related to every icon or figure depiction in the architecture. Database linking can help track architecture characteristics such as item name, cost, manufacturer, warranty status, operating system version, component size, user name, applications, etc. The linking of icons or objects with textual information provides a complete, documented view of the network and system components. The architecture can be accurately portrayed: its components, locations, functionality, protocols, and interoperability. The database of related items becomes a powerful tool for planners, engineers, and managers to conduct queries and analyses against. Standard and ad-hoc reports can be prepared to identify certain configurations and user capabilities. The database can also support accounting information for audit controls and inventories. Outputs can be generated specific to site, function, or user in order to identify assets, configuration, status, and location.

The maintenance and updating of an electronic automated architecture will also assist end users and program managers with proper configuration management and control. If you were just tasked to keep track of project cost, briefing status, or an update of maintenance problems, how would you respond? Automated tools can help. They can track the various versions of both commercial and custom-developed applications, operating systems, port configurations, and services offered. This provides a common view and reporting mechanism for all sites, highlighting software compatibilities, current status, and capabilities with regard to various implementation stages for GCCS. The automated architecture also provides a historical perspective of the program to provide insight into major upgrades, performance issues, the progress of upgrades, and migration planning for long-term evolutions such as GCCS.

Accounting and reporting functionality also supports financial management for cost data and recovery. Information stored in the database identifies costs paid, current costs, and depreciated costs. For GCCS, program managers can track total expenditures to date by site, user, component, function, and location. Additionally, the latest prices can be used to perform analysis of projected costs for expansions, upgrades, or migrations between functions and sites. The data can be exported to a standard spreadsheet for further validation and use in publishing in predefined reports.

Modeling and Simulation

Will a new segment or installation of a high-speed backbone solve current performance problems? Can a new application using the Word Wide Web (WWW) be installed on the logistics segment without adverse impact on the backbone? How can you provide a quick, clear, technically correct answer to such questions? GCCS functionality is too critical to operations to support on-site experimentation for the purposes of introducing new equipment, troubleshooting, or testing new procedures. Solutions must be accurate and unobtrusive to avoid degradation of system performance and potential impacts on mission effectiveness. To get the answers to complex questions without disrupting operations, planners need the ability to perform modeling and simulation of the architecture.

As network complexities grow and applications migrate to client/server environments, modeling and simulation tools are required more and more to properly analyze and design a new architecture. Many factors are involved in generating an appropriate design for new networks involving client/server applications and high-speed backbones. These factors include network components, protocols, vendor products, latency, buffer size, and memory. All of these factors must be considered for every aspect of a complex design to ensure that the network integrity and performance are either maintained or optimized. Modeling and simulation tools help planners and engineers to understand the complexities and interrelationships in an information system and to prevent dollars from being spent uselessly on a solution that may not work. Modeling and simulation tools help to justify program expenditures on new systems and upgrades by locating and solving problems with "what-if analyses."

Modeling and simulation tools support planners and engineers by providing performance data and network behavior patterns for both current and proposed conditions. This is key to delivering optimal performance improvements on a given budget. Will the segmentation really help, or would the addition of an Ethernet switch be a better approach? This type of question can be answered without having to procure and install expensive network components, and without having to use test centers. Modeling and simulation can also help application programmers to better understand an enterprise network and to better utilize the various protocols for testing the system design and architecture before committing to coding or modifying the application.

Modeling and simulation tools can interface with monitoring packages to gather actual data packets and measure network load helping to analyze and typify the network. The monitoring tools are used to help determine the number of packets, collisions, and processing loads on the various segments of the network. Traditionally, these tools have been used over a period of time to identify trends and help the user better understand actual network performance. Now, monitoring points and time periods are established to help examine and analyze patterns based on cycles of a day, week, or month. This information will help planners and engineers determine network behavior and predict traffic patterns for a typical day based on predictable cycles. The changes to the network with live traffic can now be better understood when considering at performance problems and potential upgrades. Some tools incorporate intelligent analysis, with automated recommendations delivered as an output product of the model. This can be as simple as "segment LAN [x] into three segments and install a new router" or "move server [f] from the Operations segment to the Logistics segment."

Modeling and simulation tools can help answer vendor-specific questions related to a design. These tools can use specific vendor information to answer questions for upgrades such as the expected performance gains for an Ethernet to FDDI upgrade. FDDI does not always deliver ten times the performance of Ethernet; it can vary from vendor to vendor. Additionally, upgrading network components can degrade network performance; adding another port on a router without adding additional memory can slow a network down.

Network and application modeling will force application designers to recognize latent network-design assumptions regarding offered load, network bandwidth, network latency, and division of labor between the client and the server. The simulation allows network designers to assess the networking costs of various application architectures and to substantiate or refute the need for costly WAN bandwidth.

The tools described in this section give designers insight into network processes and establish the network's operating parameters (similar to the use of wind-tunnel testing in aircraft design). Organizations creating client/server applications or porting applications to a WAN backbone will benefit from simulation. Additionally, users with networks that experience regular, incremental growth can use these tools to plan network redesign and to identify future bottlenecks before users notice degraded performance.

The tool sets initially examined and recommended for GCCS (and for which reports appear in Appendix C) have a limited capability in modeling and simulation. A full set of the tools identified will be available for the GCCS Engineer. Users are encouraged to test-drive these tools and determine the benefits for their own use, as some of these tools are best used for some architectures over others, vary in expense and require different user skills.

> Automated tools support three primary aspects of architecture development: documentation; accounting and reporting; modeling and simulation. The use of automated tools provides the ability to rapidly assess, analyze, and support sound decisions for future networking. The complexities and cost of networking and information systems components are beyond the “buy and try it” philosophy used in the past. With increased user demands and reduced budgets, automated tools can help prevent an improper or potentially useless upgrade.

DIFFERENT TOOLS FOR DIFFERENT ROLES

The architecture development and documentation process involves several different functional roles, as discussed in Section 5. Each of the “key players” may have a distinct set of needs for using a tool and a different view of the network. This view can range from a detailed router configuration to a C4I for the Warrior representation. The required information is dependent on the role of each player. There is, however, a common framework of information that is needed by all GCCS personnel. This section details how the various types of GCCS personnel would use architecture tools to support their missions and roles in the development process.

End User

End users, which include operators, site managers, and "power users," are concerned with the physical and functional layouts of a system. The end users provide the most important input into the system development process, as the system is built for their use. Their continuing feedback and comments are critical for a successful design. Over the years, it has been recognized that many users provide helpful and extremely detailed inputs because they have a better grasp of system requirements and installation configuration than anyone else. To assist these users, automated tools can provide a view of their local site and the installation configuration. The end users of today are computer savvy enough to help troubleshoot and to provide solid technical feedback related to application performance and satisfaction.

End users should become familiar with the tool packages listed in Appendix C for future use in developing GCCS architectures. The remainder of this section provides some brief examples of the benefits that are derived by using automated tools.

End users are interested in automated tools that support the functional, technical, design/installation/implementation, communications, and geographical architectures. Typically, end users are most interested in the documentation and the accounting and reporting functionality of the architectural process. The documentation function helps users identify and track the configuration for the local site and any other areas for which the end user is responsible. Automated tools can help end users and site managers conduct troubleshooting and fault isolation procedures using the technical architecture and installation diagrams. Many end users can use architecture information to help understand system set-ups, and to determine where needed information may reside if a problem arises.

End users are also interested in automated tools that support the functional architecture. The end users need to be able to view the functional architecture to better understand the current system functions and their interrelationships with GCCS. Automated tools will provide them with the ability to identify additional activities and functions that must be supported by future iterations of GCCS. End users are also important in developing the future architectures to the extent that such architectures relate to the local site interfaces and legacy system application support. The accuracy and functional information in future-oriented architectures can be verified by the end users to help ensure data integrity throughout the system.

One other area of support for end users is the ability to present information visually to local commanders. End users can export information and diagrams from the tools for use in presentations or reports, as needed. An export feature is critical to enable end users to use their own existing or preferred presentation packages (e.g., Harvard Graphics, PowerPoint). Today, audiences of presentations have become accustomed to presentations with more of a morphological flavor. The intent is to support the exportation of information to these packages in a Windows-standard format to help end users continue with their creative presentations while reusing actual information from the architecture.

GCCS Architect

The GCCS Architect is the central point of contact for all architecture development and maintenance activities. The GCCS Architect serves as a liaison between the technical community and the end users to ensure the system meets user requirements and operates with optimal performance.

The GCCS Architect is the historian and futurist of the development process for GCCS. The GCCS Architect is principally interested in network diagramming, presentation, and some levels of modeling and simulation. The GCCS Architect is interested in automated tools from the standpoint of longevity, stability, low cost, and ease of use.

The GCCS Architect is interested in automated tools that support features for recording diagrams and that support a method for formal acceptance and approval of architectures for GCCS. Export and import features are important, offering the flexibility to use standard office automation tools to further disseminate architecture depictions. Importation is important because it supports the GCCS Architect's receipt of information from personnel at the GCCS sites.

The GCCS Architect is in charge of the technical architecture, and therefore must be equipped with the proper automated tools to support the development and documentation of current and future configurations. Drill-down capability from a C4I for the Warrior perspective to a region or site is required. For example, the GCCS Architect can use tools to determine the feasibility of introducing ATM technology into the technical architecture. Will a LAN upgrade to ATM boost performance, or should we focus on upgrading the WAN? Should ATM be implemented on the backbone? The example in Figure 6-2 shows a diagram with an ATM switch replacing the router. This figure could be used by the GCCS Architect to include in briefings on a proposed solution. In this scenario, modeling and simulation tools may be used to help answer questions and support discussions and presentations among engineers, architects and managers.

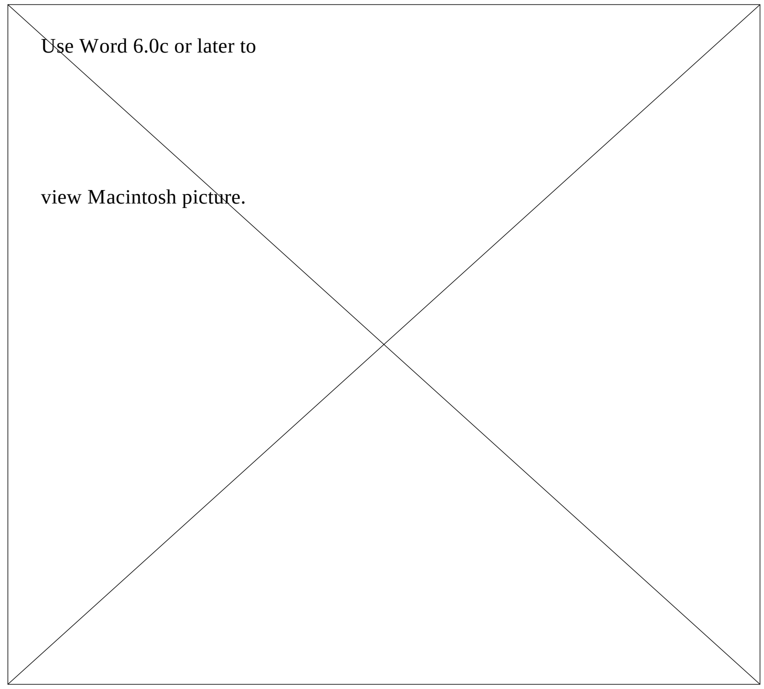

Figure 6-2.
Proposed Network Upgrade

Other questions that can be posed to the GCCS Architect may include: "Does Defense Information System Network (DISN) support ATM yet? If not, when? How much will each segment cost? What is the total cost? What if it is spread out over four years – are there any other impacts to the planned upgrades? What, in the next three years, is dependent on ATM being in place?" These are questions that can be answered when using the proper tools and automation capabilities that provide access to the information required to baseline the configuration, track improvements, identify a target architecture, incorporate database functionality, and apply modeling and simulation features.

GCCS Engineer

The GCCS Engineer validates and designs all aspects of the architecture for technical accuracy and feasibility. The GCCS Engineer also ensures interoperability with other systems and ensure that system integrity and performance are maintained as the system migrates. To accomplish these functions, the Engineer will need a full suite of architecture development tools. Knowledge of the baseline, technical, functional, target, and implementation architectures are very important for all engineers involved in GCCS. Consistent information and presentation style are critical to ensure a common framework between the GCCS Architect, the GCCS Engineer, the Program Manager, the Service Proponents and site personnel.

Engineers will maintain copies of most of the architectures and will also develop many of them as well. Import and export features for the tools are important to facilitate data exchange. Engineers will receive detailed drawings and architectures from the end users. Engineers will also receive functional and reference architectures from program managers and policy-makers. Engineers in turn will translate system requirements from the Architect into a specific technical design.

Drill-down capability is extremely important, as engineers will need diagrams of the overall network, as well as thematic layers for various functions. Engineers will also need diagrams that represent the actual site configurations, to include a drill-down capability for a rack configuration. Such information becomes important when a new segment or system is being fielded to support a new requirement. For example, if a remote capability must be supported in order to satisfy an urgent requirement, engineers can quickly examine the current equipment configuration, such as is illustrated in Figure 6-3, and determine which sites have available equipment to support the addition of remote dial-in services. Engineers can then "drill down," as is illustrated in Figure 6-4, to examine the site configurations and determine the current equipment layout and the capabilities necessary to support the new requirements. If, for example, seven new connections are required, the engineer can determine the number of ports available, the number of new ports required, the equipment needed, and whether the rack can accommodate the new equipment. This information becomes essential during crisis activities when solutions must be fielded or the components reconfigured virtually instantaneously.

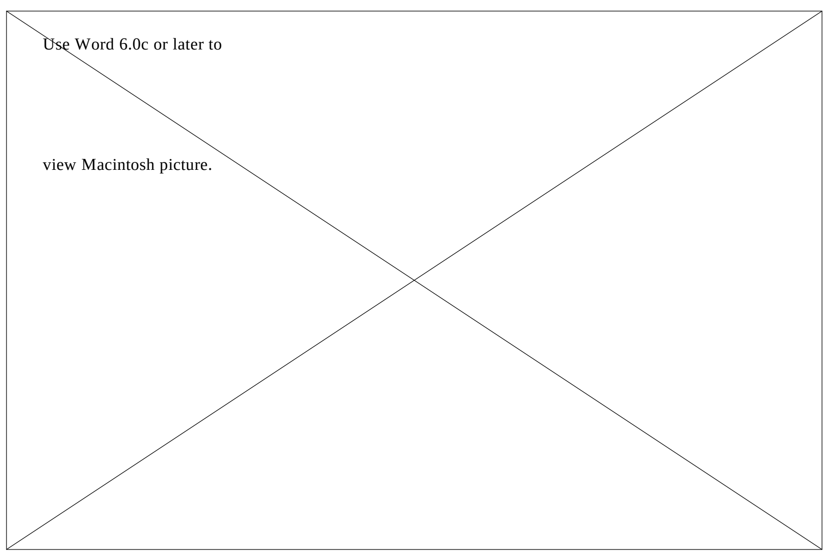

Figure 6-3.
Supporting Network Connections

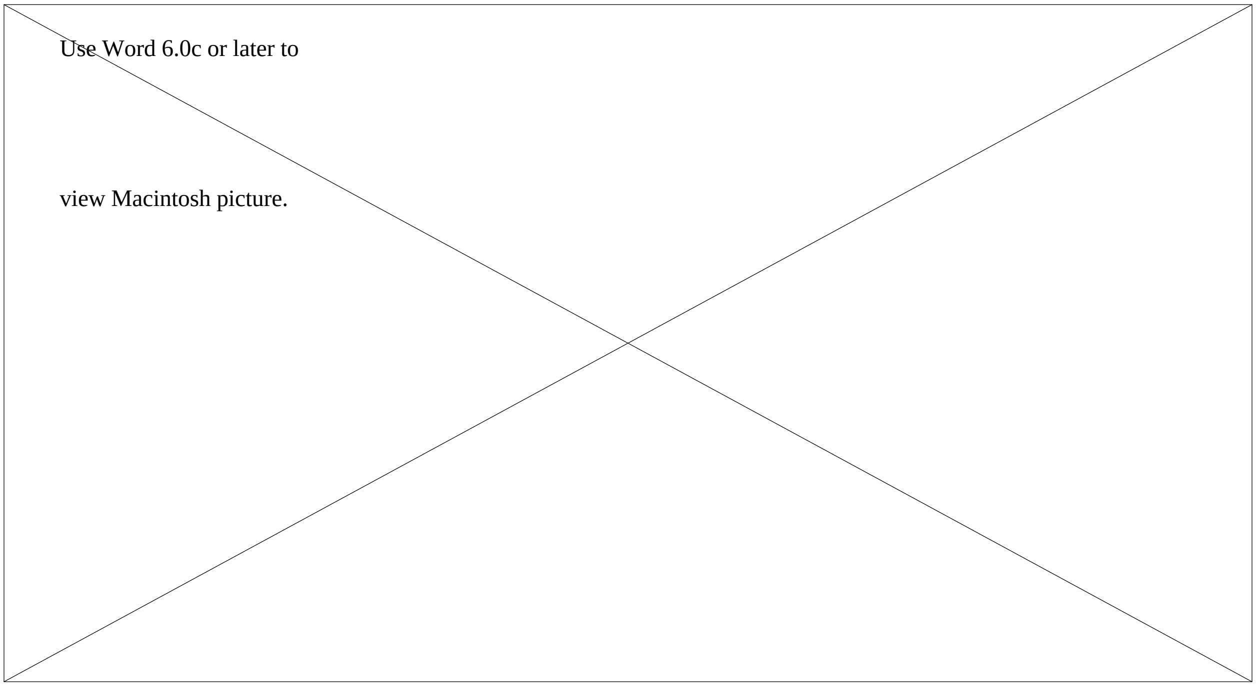

Figure 6-4.
Drill Down Capability

Engineers will use the modeling and simulation tools to help solve performance problems while examining planned upgrades. Many commercial companies operate on the principle of “buy more.” When the network clogs or when performance suffers, they just buy more equipment and install it. This “buy more” philosophy works when the reaction time and components (i.e., PCs) have a low threshold level of importance. In GCCS this philosophy will not work because of the critical support GCCS provides to the Warrior, the complexities of the hardware, and budgetary constraints.

Using automated tools, engineers can examine potential upgrade solutions without adhering to the "buy it and try it" philosophy. Modeling and simulation tools can help the engineers decide if a new application will require a new server or a new segment of the network. Will the new application run in Microsoft Windows? Will it run as an X Windows application? What will be the impacts on the local segment? What will be the impacts on the WAN? Can the router handle the new application or will new cache memory be required? Has the application been designed with the network in mind? These are typical "what-ifs" that the engineer can better answer with the proper architecture developmentand analysistools in hand.

GCCS Program Manager

The GCCS Program Manager is involved in the development and management of all programmatic strategies and financial aspects of GCCS. The Program Manager also oversees architecture implementation efforts to help assess programmatic delays, manpower, status, and financial impacts for all GCCS migration phases.

The GCCS Program Manager will use the high-level baseline, target, functional, technical, and design/implementation/ installation architectures, as well as the strategic and reference architectures. The Program Manager will need flexible capabilities for drawing relationships and generating organizational charts, in addition to having flexible import capabilities. If the program management organization uses Microsoft Organizer, the architecture tools must be flexible enough to import these diagrams into the master data sets.

The program management organization will also use architecture tools for the review of information and diagrams that are created by other personnel. The program manager can review a series of technical architecture diagrams to determine a new installation manpower assignment. By examining the architecture diagrams, the Program Manager will be able to determine the number of UNIX-based servers that are at the site or under the responsibility of the local command. From this review, the Program Manager can compare installed sites and determine the manpower requirements needed for the new site. Manpower or billets for each server's support can than be promulgated along with implementation and installation plans developed by the engineers.

The Program Manager will also use the financial aspects of the database to determine programmatic update costs. The database in the tool set can track the components, installation date, versions, initial cost, current cost, and other related figures. The tool sets support the export of data to standard spreadsheets such as Excel that can be used by the Program Manager to support financial analyses. How old are the various components at each site? What will it cost to buy new ones all at once? Can the buy be phased in over two years? Has the cost decreased or increased in the last year? Is it worth upgrading, or replacing the components? What is the trade-off point? Additionally, more complex financial modeling tools are available for the program manager to help fine-tune the financial models as part of the architecture development process.

GCCS Project Officer

The GCCS project officer manages a set or series of specific activities or components for the GCCS. The project officer oversees various aspects of the architecture implementation process and programmatic implementation activities. These implementation areas can include issues pertaining to routers, servers, specific application servers, or specific sites.

The project officer is interested in baseline, target, functional, technical, design, installation, and implementation architectures, and will also use the communications/ hierarchical information and reference architectures. The project officer will need tools that can support drill-down features to track specific diagrams and activities. For a router manager, the ability to track the installed configuration, rack layout, interfaces, port assignments, and software uploads is necessary. This information can be provided by the end user's site manager, the engineers, or the project officer. The import and export features are important. The project officer will be using a variety of office automation tools and will support a wide spectrum of DoD users as part of GCCS. The architecture tools must be flexible to support the project officer's existing application suite and to provide the capability to import and export information with minimal impact or having to reenter the information. Schedules, planning figures, detailed diagrams, and detailed item descriptions should be easily moved among applications to support decision-making and use of the information regardless of the level of detail.

The project officer will develop and review architectures related to their specific component or activity. In the case of the Operations/Intelligence server, a project officer may be assigned for that specific activity. This project officer would need to be aware of where all the servers are located, how many users are being supported at each site, the type of server, disk storage capability, and performance levels. Much of this information is available from the architectural drawing tools. The architectures and tools can provide answers to questions on how much it will cost to migrate the applications to a Silicon Graphics server, how much disk space is available for new imagery, if two new modules can be added without impacting server performance, if all servers have the same memory configuration, and how much it will cost to upgrade all servers to a FDDI based connection

The project officer will also be interested in gathering outputs and/or using the simulation and modeling tools when the specific activities relate to increasing or validating performance. Just as is the case for engineers and program managers, a project officer can use the modeling and simulation tools to help verify proposed tools and run analyses against potential solutions. This entails detailed and complex modeling to help review buffer issues, memory sizes, user network segments, and server domain set-ups. Project officers can examine the need to increase server memory and component memory, separate segments for back-up and recovery operations, and analyze vendor specific components. Using the vendor specifications, the project officers can input the various parameters to help determine the most cost-effective approach. Which new router should be selected? Is it dependent on the existing set? Can a 3Com router interoperate with a Cisco router using System Network Architecture (SNA)? Does Cisco provide a higher packet throughput for TCP/IP over FDDI? The tools can help the project officers best by supporting the actual architectural implementation process, mitigating a majority of the need for trial and error or guessing.

Policy Maker

The GCCS policy makers are involved in establishing the policy that drives the architecture requirements. The policy makers also devise and/or support strategies related to the overall architecture. The policy makers establish policies and objectives such as the cut-over of Worldwide Military Command and Control System (WWMCCS) or the incorporation of other systems or programs. To help reach decisions in a timely fashion, the policy makers require information on functional, technical, strategic, geographical, reference, representational, and design/implementation/installation architectures.

The policy maker will mostly use the outputs of architecture tools and will not be inputting information on a regular basis. The export feature is important for the policy maker. The export feature can provide the policy makers with high-level C4I for the Warrior-type diagrams for insertion in briefings to senior leadership in DoD and Congress. The diagrams can also be viewed to determine the overall architecture and progress made to date on migration and implementation. This can be used visually to present information for future funding, describe past efforts, and provide justifications as well.

> Automated tools are primarily intended for the GCCS Architect and Engineer. However, the tools support the complete spectrum of GCCS players as demonstrated here. All users are actively encouraged to acquire their own tools and use them as needed in supporting their roles.

COMMON TOOL REQUIREMENTS

This section identifies the GCCS architecture development and analysis tool requirements for each of the functional tool categories. This data will highlight overlaps between various packages to help identify Commercial-off-the-Shelf (COTS) products that support more than one functional tool area and may be best for use in GCCS. The functional tools have been grouped into five basic areas: network drawing, network documentation, presentation, discovery, and modeling and simulation. The features and requirements for each type of GCCS personnel are grouped together and discussed for each functional area. These functions are used in Appendix C for evaluating potential off-the-shelf packages that can be used to support GCCS.

Network Drawing

A network drawing is a Computer Aided Design (CAD) type drawing capability to capture locations and physical layouts. Networking drawing functionality required includes:

- *Drag and drop palette of tools*
- *Multilevel diagramming with navigation between levels*
- *Hide on-screen data*
- *Comprehensive set of computer and telecommunication node graphics with leading vendor specific icons*
- *Ability to design custom symbols and import symbols/icons from popular packages*
- *Ability to scan vendor brochures and import depictions as a graphic icon*
- *Capability to support flowchart and data flow symbology*
- *Ability to import background features for floor plans, buildings, cities, states, countries, and world maps*
- *Create links between icons that are straight, segmented or curved*
- *Embed attributes in or with the icon, moving or copying and bringing all data with the icon including lines connected to objects*
- *Include text labels as part of the icon graphic*
- *Ability to make engineering drawing borders.*

Network Documentation

Network documentation features are the expanded details to complete the documentation of a network such as computer types, applications running, configurations, etc. This is supported by the database behind the icons and the ability to print out the full set of information related to the network diagram. The functionality to support this is defined as follows:

- *Define, enter, view, display, import, and export object-oriented data (graphic symbols, diagrams, text)*
- *Populate diagrams from existing spreadsheets or databases*
- *Refresh data from external sources*
- *Ability to export data to another database application such as Microsoft Access*
- *Ability to import data from another database application such as Microsoft Access*

- *Ability to export data to a spreadsheet such as Microsoft Excel*
- *Capability to support built-in preformatted report development with ad-hoc reporting features*
- *Customizable data attributes and fields*
- *Relational database compliant with open database connectivity (ODBC) standards*

Presentation

This area addresses the suitability and support for visually presenting the architectures and supporting scheduled and ad-hoc briefings. The presentation functions include:

- *Combine both graphical and database information into one view*
- *Ability to create a different depiction from a common set of data, including flowcharts, briefing slides, functional blocks, wiring diagrams*
- *Zoom capability to provide the necessary details appropriate for the intended audience (i.e., high level geomorphic graphics for senior leadership and vendor specific icons for engineers and planners)*
- *Ability to import or export graphics from traditional office presentation tools such as PowerPoint or Lotus Freelance*
- *Ability to import in vector or bit-mapped images for high profile presentations*

Auto Discovery

Auto discovery is the capability to use probing agents to automatically go out and identify various components on a particular network. According to industry experts, a good autodiscovery tool can help save over 25 percent of the time to properly document a network. Autodiscovery is also important for rudimentary understanding of basic networks for rapid understanding and troubleshooting when no known documentation exists for the network. The autodiscovery features required include:

- *Ability to query network and automatic poll all SNMP agent equipped devices such as routers, hubs, gateways, and GCCS servers*

- *Ability to draw a network layout based on the Simple Network Management Protocol (SNMP) results and IP addresses received*

- *Ability to export the results both graphically and textually to other drawing and management packages*

- *Capability to interface with network management packages,*

- *Capability to perform basic Network Management functions (e.g., Ping-based diagnostics, trace route, auto survey capability, interface to SNMP Management Information Base [MIB])*

- *Based on industry and DoD standards (TCP/IP), SNMP, and MIB*

- *Low performance impact on the network and managed objects*

- *Vendor supported and part of the GCCS architecture*

Modeling and Simulation

Modeling and simulation tools are imperative to the generation of analyses and what-if scenarios. Modeling and simulation can assist both planners and administrators in providing and verifying the optimal solution for a network or application. The modeling and simulation tool requirements are defined as:

- *Ability to provide accurate data and ease of entering the data from the existing drawing packages or tools including network devices, configuration, and traffic*

- *Ability to read in live network traffic from an analyzer*

- *Capability to model specific GCCS components and equipment such as SUN and Cisco*

- *Access to libraries of predefined modules for networking devices, protocols, and topologies*

- *Ability to customize the network data traffic for specific analysis and configurations for GCCS*

- *Analysis capability to conduct what if scenarios at various levels*

- *Ability to provide near real-time configuration data for each GCCS site*
- *Ability to provide financial modeling for operating and equipment cost*
- *Capability to support new and future networking equipment for new WAN technologies and high speed services such as ATM.*

TYPES OF TOOLS AVAILABLE

The types of tools available to support the architecture development and documentation process encompass three main areas: network design, auto discovery, and modeling and simulation. Three of the five identified areas of functionality namely network diagramming, network documentation and presentation, are inherent in the network design functionality. Most packages today support all three functional areas to some extent. Any package that was solely for one specific functional area (e.g., PCPaint, PowerPoint, or Paradox) was not considered an architectural tool appropriate for supporting the GCCS process. A generic drawing tool (e.g., Micrographix's Designer and Corel's CorelDraw) can be used for connecting clip-art and generating consistency among hardware illustrations for network diagrams. However, these packages offer over 10,000 accurately depicted clip-art images, of which only a small number relate to network design and architectures. These packages are important for users to import or export information from, but are not considered part of the primary architectural tool set.

The types of tools identified for GCCS are representative of the COTS tool sets found in industry. This sampling is not all-inclusive, and users are encouraged to identify other offerings. Typical packages that can meet identified needs are detailed in Appendix C.

!

The list presented in Appendix C is not to be used as a buying guide. It is included to give you an illustration of some of the products and packages available in the market. The following lists also illustrate key functionalities associated with popular network diagramming tools.

RECOMMENDED TOOLS FOR GCCS

During the course of the development of this report, it was recognized that no one tool can meet the complete needs of the GCCS architecture methodology and documentation process. However, commercially available products do meet many of the basic requirements in one form or another, and combinations of tools can provide much of the functionality needed today. Various tool sets have been proposed and evaluated for use in GCCS. The list of the packages and tools, along with the results of the evaluation appear in Appendix C. New tools and applications are continually being introduced. You are always encouraged to identify useful tool sets and packages to the GCCS Architect for further consideration and inclusion in future iterations of Appendix C.